UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería de la Tecnología y Servicios de las Telecomunicaciones

TRABAJO FIN DE GRADO

Procesamiento de Texto Manuscrito I: Segmentación a nivel de palabras, indexación y clustering

Nerea Romera Vicente
Tutor: José Colás Pasamontes

Julio 2017

Procesamiento de Texto Manuscrito I: Segmentación a nivel de palabras, indexación y clustering

AUTOR: Nerea Romera Vicente
TUTOR: José Colás Pasamontes

HCTLab
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2017

Resumen

Este Trabajo de Fin de Grado tiene como objetivo proponer una solución que permita la segmentación e indexación de palabras extraídas de textos manuscritos. La motivación de este proyecto surge dada la escasa investigación que existe sobre el tema. En la actualidad, el único proceso de digitalización que se ha llevado a cabo con algunos textos manuscritos es escaneado/fotografía de los mismos con ayuda de escáneres occipitales que respetan las condiciones naturales de los libros. Este proceso de digitalización se realizó con la premisa de poner a disposición de expertos en la materia todo el contenido histórico para facilitar su análisis. No obstante, a pesar del cambio de soporte, el proceso sigue siendo totalmente manual. Por tanto, se considera este proyecto como una oportunidad para poner la tecnología a disposición de la historia, creando una herramienta que tenga la capacidad de mejorar las características de las imágenes, a través de una fase de pre-procesamiento, para posteriormente efectuar una segmentación de los textos en líneas y palabras. Los resultados de la segmentación se guardaran en una base de datos que contendrá un archivo de imagen, así como una referencia a la ubicación de las palabras (acta de la que provienen, línea en la que se sitúa, y posición en la línea). Con esto, se espera que en trabajos posteriores se alimente la herramienta de un algoritmo dinámico de agrupamiento para que el tiempo de trabajo de paleógrafos e historiadores disminuya notablemente.

Palabras clave

Binarización, *Bounding Box, CCL*, clustering, etiqueta, filtro, histograma, manuscrito, preproceso, RGB, ruido, segmentación, *skew,* subdominios

Abstract

This End-of-Grade Paper aims to propose a solution that allows the segmentation and indexing of words extracted from handwritten texts. The motivation of this project arises from the scarce research that exists on the subject. At present, the only process of digitization that has been carried out with some handwritten texts is scanned / photographed with the help of occipital scanners that respect the natural conditions of the books. This digitization process was carried out with the premise of making available to experts in the field all the historical content to facilitate their analysis. However, despite the change of support, the process remains entirely manual. Therefore, this project is considered as an opportunity to make the technology available to History, creating a tool that has the ability to improve the characteristics of the images, through a pre-processing phase, to later carry out a segmentation of the texts in lines and words. The results of the segmentation will be saved in a database containing an image file, as well as a reference to the location of the words (record from which they come, line in which it is situated, and position in the line). With this, it is expected that in later works the tool of a dynamic grouping algorithm will be fed so that the working time of paleographers and historians will decrease significantly

Keywords

Binarization, Bounding Box, CCL, clustering, label, filter, histogram, handwritten, preprocesing, RGB, noise, segmentation, skew, subdomain

Agradecimientos

A mi tutor, José Colás, por su paciencia y dedicación.

A mis padres y heraman, por su confianza y apoyo constante.

A Sergio, el amor de mi vida, por darme tanto y no pedir nada a cambio.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

1 Introducción

1.1 Motivación

El origen de los textos manuscritos se remonta a la época del Antiguo Egipto. Desde entonces, a lo largo de la historia el número de manuscritos ha ido incrementándose de manera exponencial. Por desgracia, muchos de ellos no han perdurado y se ha perdido la oportunidad de realizar estudios paleográficos, debido a su baja calidad tipográfica [1] por su desgaste con el paso del tiempo.

Para contextualizar el ámbito de este trabajo y sus motivaciones, es de especial importancia definir qué es la paleografía. La paleografía [1] es la ciencia que se encarga de descifrar las escrituras antiguas y estudiar su evolución, así como datar, localizar y clasificar los diferentes testimonios gráficos objeto de estudio. Es objeto de la paleografía el examen crítico y sistemático de los elementos gráficos de la escritura, forma alfabética, signos accesorios, abreviaciones, notas musicales, reconocimiento de mano, correcciones del copista o de los editores. El paleógrafo debe dominar bien la lengua de los textos y sus particularidades gráficas, o sea, los estilos, las abreviaturas y los anagramas, ligogramas y nexogramas, entre otras. Dichos conocimientos son esenciales para que el paleógrafo pueda descifrar el texto antiguo, así como asignarle una fecha y un lugar de origen.

Dada la dificultad que los estudios paleográficos conllevan, y teniendo en cuenta que la pérdida de todos estos textos supone un riesgo para la consolidación histórica, surge la necesidad, y motivación principal de este trabajo, de encontrar una solución que en primer lugar permita digitalizar el contenido histórico para asegurar su preservación; en segundo lugar definir un método que facilite las labores paleográficas, siendo agnóstico a la lengua en la que esté escrito y al origen de la fuente histórica.

Así mismo, otro hecho que motiva al estudio de la materia es el decremento notable de profesionales en el tema. Con el paso del tiempo la paleografía ha dejado de impartirse como contenido obligatorio en las universidades, y ha quedado en un segundo plano, lo que ha provocado que el número de especialistas disminuya, y que exista la posibilidad de que puedan desaparecer.

1.2 Objetivos

El objetivo de este trabajo es poner a disposición de los profesionales de la paleografía, una herramienta que saque el máximo partido a la capacidad de cómputo que ofrece hoy en día la tecnología, cuya función sea la extracción de palabras para un posterior agrupamiento en grupos de palabras similares.

Esta herramienta ha de permitir digitalizar y segmentar el contenido histórico de estos textos para facilitar la transcripción de los mismos, y de este modo contribuir a su preservación y estudio a lo largo del tiempo. Para ello se definen tres objetivos principales que han de ser cubiertos para el correcto funcionamiento de la herramienta.

El primer objetivo es diseñar un módulo de preprocesamiento cuya función sea eliminar todas las imperfecciones que puedan afectar al posterior tratado de la imagen y que desemboque en la generación de un resultado no satisfactorio.

El segundo objetivo es la segmentación de los textos en líneas y palabras a través de un algoritmo robusto que permita obtener resultados óptimos, independientemente de la naturaleza de la muestra y de todas aquellas imperfecciones que no hayan podido ser eliminadas en la fase de preprocesamiento.

El tercer y último objetivo es etiquetar todas las palabras extraídas, haciendo referencia a la página, línea, y a la posición que ocupan dentro de la misma, con el fin de tener una referencia en la propia imagen que permita volver en cualquier momento al contexto en el que es mencionada.

En definitiva, esta herramienta ha de permitir a un transcriptor de textos históricos ser más eficiente a la hora de llevar a cabo su trabajo, ya que únicamente tendrá que centrarse en la transcripción de muestras individuales, y por tanto un alto porcentaje de su trabajo esté automatizado disminuyendo notablemente el tiempo de estudio.

1.3 Organización de la memoria

La memoria se organiza en los siguientes capítulos:

- En la primera sección, se describe la problemática y motivación que se ha tenido para abordar este Trabajo de Fin de Grado. Así como los objetivos definidos para la resolución del problema.
- En la segunda sección, se hace mención al estado del arte.
- En la tercera sección, se describen con detalle los módulos que componen la herramienta diseñada.
- En la cuarta sección, se explican minuciosamente los procesos seguidos para el desarrollo de la solución.
- En la sección 5, se habla sobre la muestra con la que se han realizado las pruebas, y los resultados obtenidos.
- Por último, en la sección 6, se ponen en manifestó las conclusiones extraídas, así como se habla del trabajo futuro de este proyecto.

2 Estado del arte

En esta sección se va a realizar un recorrido a través del estado del arte para las técnicas empleadas en la resolución de este problema. Para ello se abordará cada uno de los objetivos planteados anteriormente, y por cada uno de ellos se explicará qué métodos y técnicas se han estado usando para la resolución del problema planteado.

2.1 Preprocesamiento de la imagen

El preprocesamiento engloba múltiples técnicas cuyo objetivo es mejorar notablemente la calidad de las imágenes. En primer lugar, una de las características que entorpecen [2] el procesamiento de imágenes es la diferenciación de objetos dentro de una imagen. Este hecho se produce ya que la mayor parte de los formatos de imágenes digitales son muy pesados, y contienen mucha información que no aporta nada al procesamiento de los textos. La primera solución que se encuentra dentro de la fase de pre-proceso de imágenes es la binarización.

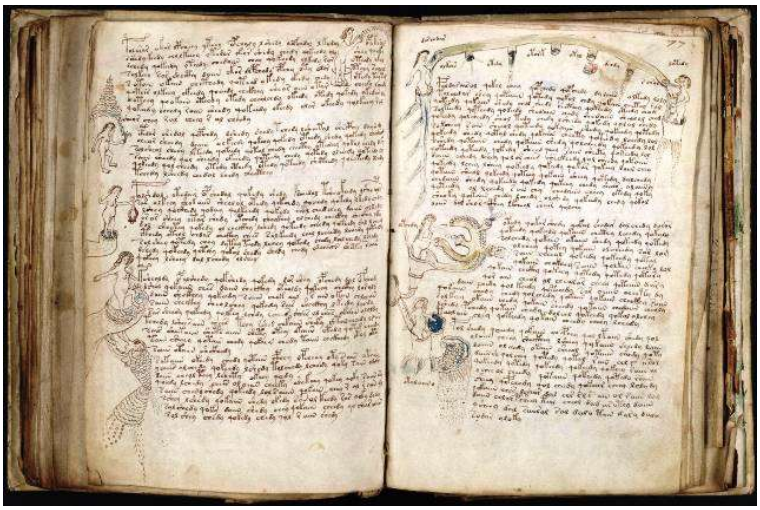

Ilustración 1. Libro manuscrito

En el proceso de binarización nos encontramos múltiples retos para mejorar la calidad de la imagen. Multitud de incunables cuentan con infinidad de imperfecciones, tal y como el ruido, manchas, inclinación del texto, fondos desgastados, caligrafías usadas, etc. Todos estos defectos suponen una problemática para el procesamiento (segmentación) de las imágenes, ya que prácticamente imposibilitan el proceso. Por tanto, dentro del procesamiento de imágenes, existe una fase preparatoria que es esencial para obtener buenos resultados. Esta fase es el pre-procesamiento.

La binarización de imágenes es el proceso por el cual se busca un umbral óptimo que permita distinguir en una imagen los objetos del fondo de los objetos del primer plano. Este umbral es el punto (o valor) en el cual el histograma de una imagen se divide en dos picos. En la mayoría de las imágenes este valor resulta un poco difícil de encontrar gráficamente debido a la complejidad de estos histogramas. Es por eso que se usan métodos paramétricos y no paramétricos que modelizan el problema y encuentran diferentes maneras de obtener este umbral. Existen distintos métodos [2]:

- **Métodos basados en la forma del histograma:** Esta clasificación abarca las diferentes propiedades de un histograma, como por ejemplo los picos, valles y curvaturas.
- **Métodos basados en la clusterización:** Son aquellos que modelan el histograma como una superposición de funciones gaussianas.
- **Métodos basados en la entropía:** Usan la entropía de los niveles de gris en una imagen. La máxima entropía es interpretada como la máxima información transferida y es el umbral óptimo a elegir.
- **Métodos basados en los atributos de la imagen:** Consisten en técnicas que seleccionan un valor de umbral t basado en atributos que buscan una medida de similitud entre la imagen original y la imagen binarizada. Estos atributos pueden ser: bordes, formas, momentos de niveles de gris, conectividad, textura o estabilidad de los objetos segmentados.
- **Métodos basados en información espacial:** A diferencia de los métodos anteriores que utilizan el valor de gris de cada píxel, estos los algoritmos dependen de la información espacial de los píxeles, por ejemplo las probabilidades de su contexto, funciones correlación, probabilidades de coocurrencia, modelos locales dependientes de píxeles, entropía en bidimensional, etc.
- **Métodos basados en características locales:** adaptan el umbral en cada píxel en función de las características locales de la imagen tales como rango, varianza, parámetros de superficie.

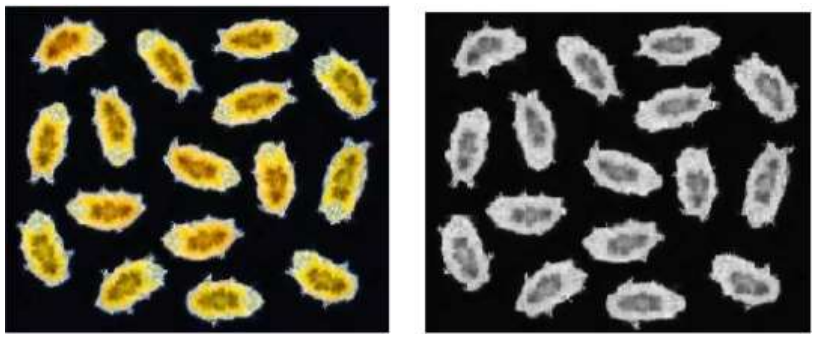

Ilustración 2. Imagenes RGB y escala de grises

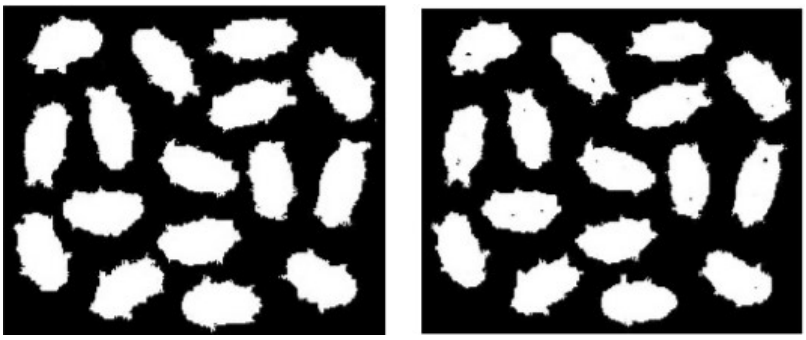

Ilustración 3. Imágenes binarizadas métodos basados en la entropía

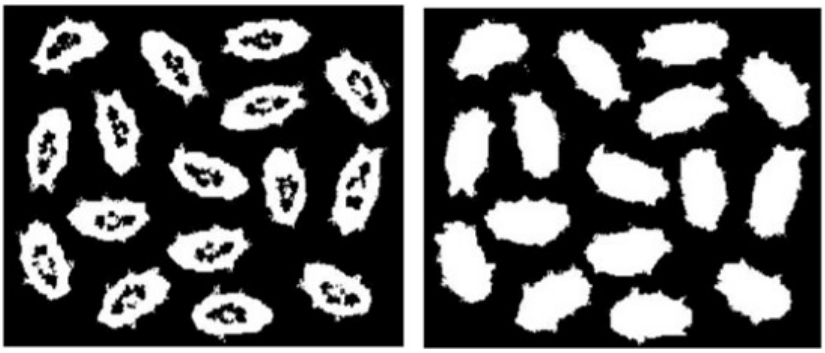

Ilustración 4. Imágenes binarizadas con métodos de clustering

Otro problema típico en la transcripción, es la existencia de desencuadre o skew. El skew [3] es una forma de ruido introducido al escanear el documento, y consiste en la falta de alineamiento del documento de papel con respecto a las coordenadas del escáner utilizado para su digitalización. La corrección del desencuadre facilita la extracción de párrafos y líneas de texto de los documentos. La extracción de frases a partir del texto es muy difícil, sino imposible, si no se reconoce previamente el texto, por lo que el módulo segmentador suele devolver el texto segmentado por líneas, sin tener en cuenta si corresponden a una frase, a parte de una frase, o a varias frases.
En líneas generales la corrección del skew se reconoce como la rotación de una imagen hasta que la posición de las líneas coincide con el eje de abscisa. Existen múltiples técnicas para la corrección del skew:

- **Basada en proyecciones horizontales:** Este tipo de aproximaciones parten de la suposición de que los documentos tienen el texto dispuesto a lo largo de líneas paralelas, cosa que ocurre en la gran mayoría de los documentos. Estos métodos son considerados los más rápidos y son simples de implementar.

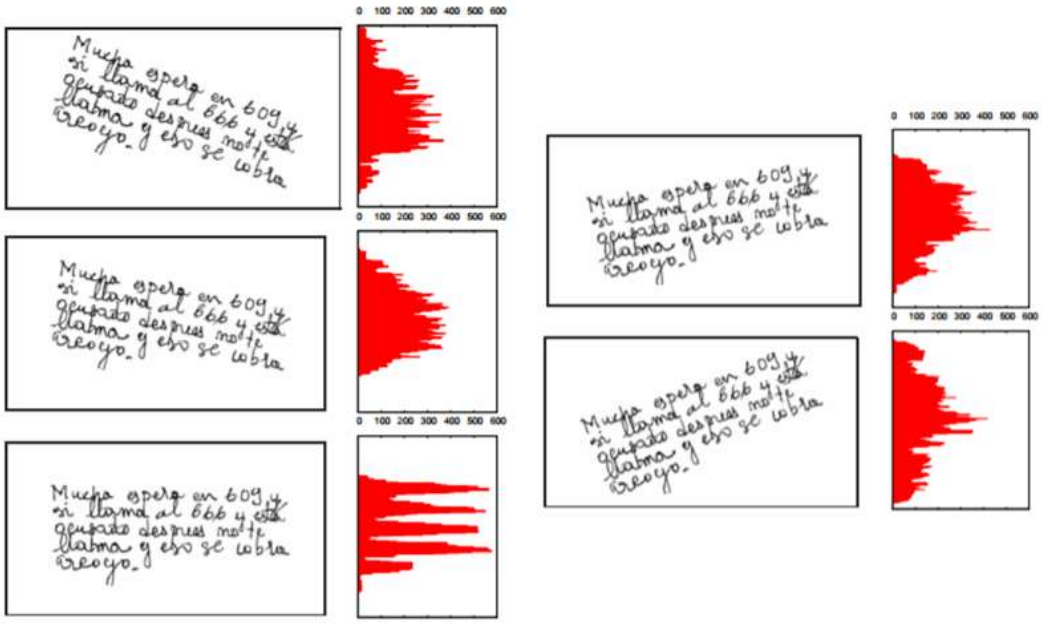

Ilustración 5. Proyección horizontal de texto con skew

Como se aprecia en las proyecciones, cuanto menos desencuadre presenta un párrafo, su proyección presenta picos más grandes y valles más profundos.

- **Basada en la transformada de Hough:** Estos métodos utilizan la transformada estándar de Hough que traslada las coordenadas cartesianas (x,y) de cada píxel negro al dominio polar (ρ,θ).

$$\rho = x\,cos\theta + y\,sin\theta$$

Ilustración 6. Ecuación Transformada de Hough

Todos los puntos (x,y) en el espacio cartesiano que estén alineados a lo largo de los máximos locales en el acumulador de Hough H(ρ,θ) representan líneas rectas, o lo que es lo mismo, representa las líneas rectas para las cuales tenemos más puntos en el dominio cartesiano. El máximo absoluto en H(ρ,θ) corresponderá a la línea con mayor número de puntos en la imagen original.

Los métodos basados en la transformada de Hough tienen el inconveniente de que son lentos debido a que deben aplicar la transformada de Hough a todos los píxeles negros y para todo el rango de ángulos a detectar una recta.

- **Basada en clustering de los vecinos más cercanos:** Estos métodos suelen empezar con un proceso de etiquetado de componentes conexas de la imagen. Se requieren imágenes binarias, por lo que si la imagen fue adquirida con niveles de gris, hay que umbralizarla previamente. Para evitar ruido, y eliminar figuras y

bordes negros, todas las componentes que sean mucho más grandes, o mucho más pequeñas que la media de componentes, se eliminan. A partir de aquí, se van agrupando bloques con características similares formando componentes más grandes. Finalmente se intenta estimar el ángulo de desencuadre para estas componentes. Hay que decir que cuanto mayor número d componentes conexas mayor es la precisión del método.

2.2 Segmentación de líneas y palabras

El campo de estudio de la segmentación de palabras está estrechamente ligado con la fase anterior, ya que si una imagen no se somete a un pre-procesamiento exhaustivo, la segmentación va a ser una tarea prácticamente imposible.

La segmentación de una página de texto requiere un análisis del su contenido. Los sistemas de análisis de página intentan obtener una representación jerárquica de la misma, donde cada bloque representa una zona homogénea de la página: una imagen, una columna, cabecera de texto, etc.

Para la segmentación de palabras deben ser considerados estos tres factores [4]:

- Similitud: los píxeles pueden tener características similares (nivel de gris, color, borde,…)
- Conectividad: los objetos se corresponden áreas de píxeles conectadas.
- Discontinuidad: Los contornos delimitan unos objetos de otros.

La mayoría de los sistemas de análisis de página se pueden clasificar en: basados en componentes conexas, basados en RLSA o basados en proyecciones.

Prácticamente todos los métodos son combinaciones de múltiples técnicas y algoritmos.

Un método utilizado para segmentar los bloques de texto está basado en proyecciones horizontales [3]. Los valles de estas proyecciones son posibles puntos de segmentación. Si un valle tiene un valor de cero, representa una zona de la imagen que no contiene texto, y por esa razón es descartada y sus fronteras son tomadas como candidatos a puntos de segmentación. Si los ascendentes y los descendentes de dos líneas se cruzan o tocan, provocan valles con valores superiores a cero. Se utiliza un umbral ρ a partir del cual los valles delimitarán las zonas de texto.

Otro método existente para la segmentación de líneas y palabras está basado en la Transformada de Hough [4] [7]. Este método es un proceso más lento, pero más robusto que el anterior, ya que es capaz de identificar líneas y palabras aunque exista skew en los textos a segmentar.

Está técnica se combina con el algoritmo *Connected-Component Labeling,* tanto en la fase de pre-procesamiento, como de post-procesamiento. En base a la experimentación, se definen tres subdominos en función del tamaño de los componentes conectados, que permiten determinar que componentes son válidos para aplicar la transformada de Hough, y cuales introducen error al detectar las líneas.

Por último, otro método es el algoritmo *Run Length Smoothing Algorithm* (RLSA) [5]. Este es un método que se puede utilizar para la segmentación de bloques y la discriminación de texto. El método desarrollado para el Sistema de Análisis de Documentos consta de dos pasos. En primer lugar, un procedimiento de segmentación subdivide el área de un documento en regiones (bloques), cada una de las cuales debe contener sólo un tipo de datos (texto, gráfico, imagen de semitonos, etc.). A continuación, se calculan algunas características básicas de estos bloques.

El RLSA básico se aplica sobre una imagen binaria, en el caso de Matlab, representada por 0 en caso de píxeles negros y 1 en píxeles blancos. El algoritmo transforma la secuencia binaria de entrada en otra de salida en base a estos argumentos:

- Se determina de manera manual un umbral C.
- Si la secuencia de entrada contiene un píxel blanco, este se modifica a negro siempre y cuando el número de píxeles blancos adyacentes sea menor o igual que el límite predefinido C
- Si en la secuencia binaria de entrada hay un pixel negro este no se modifica en la secuencia de salida del algoritmo.

Este proceso se realiza fila por fila, y columna por columna especificando para cada proceso un umbral C distinto. Aplicar el algoritmo por separado para filas y columnas, genera dos secuencias distintas. En este algoritmo las dos secuencias se combinan utilizando la operación lógica AND. Este suavizado horizontal produce el resultado de la segmentación lineal.

Ilustración 7. Imagen segmentada con algoritmo de proyección horizontal

Ilustración 8. Imagen segmentada con Transformada de Hough

Ilustración 9. Imagen segmentada con algoritmo RLSA

2.3 Indexación de palabras

La indexación se define como la ordenación de una serie de datos o informaciones de acuerdo con un criterio común a todos ellos, para facilitar su consulta y análisis.
En el ámbito de este Trabajo de Fin de Grado, la indexación es un proceso automático por el que a cada imagen se le asignan unos índices que facilitaran su búsqueda en el texto original.

En otros ámbitos, la indexación es una potente herramienta utilizada en distintas bases de datos, ya que optimiza el rendimiento de las mismas.
En el ámbito de las páginas web, es un tema de tendencia, ya que la indexación está estrechamente ligada con el posicionamiento de páginas web en buscadores y mejora de los motores de búsqueda.

3 Diseño

3.1 Componentes de la solución

A lo largo de esta sección se definirán los módulos que componen la herramienta desarrollada en este Trabajo de Fin de Grado. Cada uno de los módulos ofrece una solución a los objetivos propuestos para este proyecto.

Ilustración 10. Arquetipo herramienta

Las funciones de pre-procesamiento se encuentran en el primer módulo de la herramienta. Es la sección más importante, ya que en ella se definen ciertos parámetros que permitirán obtener resultados exitosos en fases posteriores. La variabilidad de las fuentes de origen hace de esta fase, un proceso crítico, que ha de ser configurado para cada caso.

El segundo componente de la herramienta, es el encargado de segmentar los textos manuscritos; en primer lugar a nivel de líneas y finalmente a nivel de palabras. Es de vital importancia la división de la segmentación en dos ciclos, ya que el objetivo principal de este trabajo va más allá de la segmentación de palabras como técnica, sino de la aplicación de la misma para el estudio y transcripción de estos textos.

El tercer bloque tiene una estrecha relación con lo anteriormente mencionado de la subdivisión en ciclos de la segmentación, ya que tiene como objetivo la indexación de las palabras. Gracias a esta parte de la herramienta, las imágenes contendrán información acerca del acta de la que provienen, línea en la que se ubican y posición dentro de la misma.

3.2 Definición de módulos

A continuación se definirán en detalle el diseño de cada uno de los módulos presentes en esta herramienta: Módulo de pre-procesamiento, módulo de segmentación y módulo de indexación.

3.2.1 Bloque 1: Pre-procesamiento de la imagen

El pre-procesamiento tiene como objetivo mejorar la calidad de la imagen a través de múltiples técnicas. A través de estas técnicas podremos eliminar ruido, suvizar la imagen, detectar bordes, etc, que facilitarán procesos posteriores, como en el caso que atañe a este proyecto, la segmentación de palabras.

Para poder llevar a cabo la fase de preproceso, en primer lugar se realiza la adquisición de las imágenes, de modo que todas sus características estén recogidas en el sistema. Estas imágenes pueden tener diferentes formatos, por lo que en base a sus dimensiones y forma se determina si la imagen tiene un formato binario, RGB, etc.

Esta herramienta se ha probado con una base de datos en la que todas las imágenes contaban con la presencia de un borde negro debido al proceso de escaneado. Con el fin de parametrizar y calibrar la herramienta se procede a la eliminación del mismo, ya que puede interferir en fases posteriores.

El siguiente paso dentro de esta sección es la binarización de la imagen. Para ello, la imagen ha sido convertida previamente a escala de grises. Esto simplificará los procesos posteriores, ya que la herramienta solo tendrá que trabajar con imágenes con formato unidimensional.

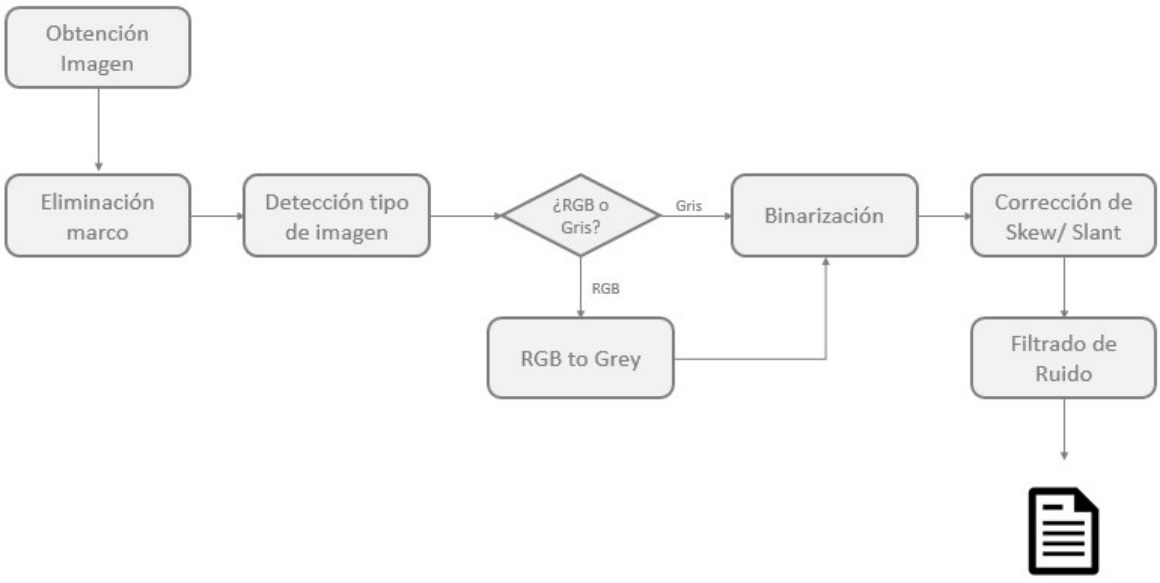

Ilustración 11. Módulo de preproceso de la imagen

Así mismo, en este bloque se podrá corregir la inclinación del texto respecto al horizonte (skew y slant).
Finalmente, la herramienta contará con una sección para el filtrado del ruido basado en la acumulación de píxeles negros entre líneas y en los bordes de las páginas.

3.2.2 Bloque 2: Segmentación a nivel de línea y palabra

En esta fase se procederá a la segmentación de palabras en los textos manuscritos, con el fin de poder generar una base de datos con información suficiente (módulo de indexación) como para poder ejecutar posteriormente un algoritmo de clustering dinámico, que busque el mayor número de coincidencias en el texto, y generando de este modo grupos de palabras iguales. De modo que se pueda reducir el tiempo de análisis y transcripción que los paleontólogos emplean en el estudio de estas piezas. Para ello el algoritmo se subdivide en dos ciclos: segmentación a nivel de líneas y segmentación a nivel de palabras.

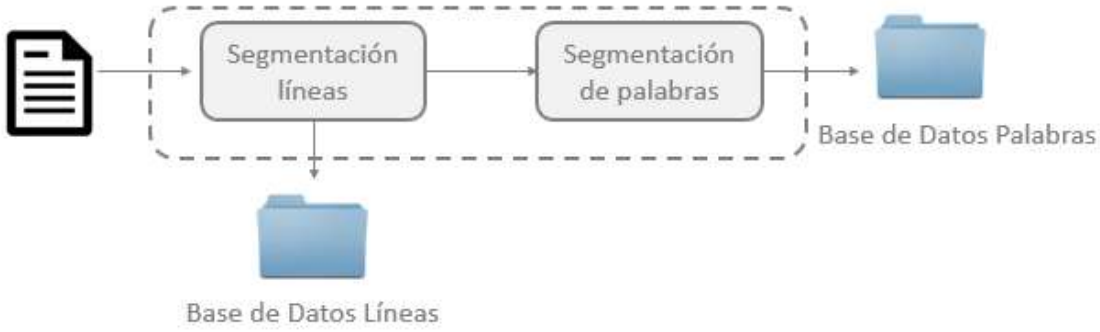

Ilustración 12. Módulo de segmentación

La operativa de esta fase es sencilla. Una vez se obtiene la imagen preprocesada, la cual tiene que tener un formato binario, y contener la menor cantidad de imperfecciones posible, se realiza una segmentación a nivel de líneas. Esta información pasará al siguiente modulo que será el encargado de generar una base datos con todas las líneas extraídas. De igual manera, esta información pasa al módulo de extracción de palabras.

El algoritmo de extracción de palabras se simplifica ya que la detección de componentes dentro del texto, se ejecuta línea a línea, de este modo partimos de una imagen más sencilla.

En una primera instancia, se intentó ejecutar esta fase partiendo de la segmentación de las líneas de la imagen preprocesada. Como los resultados no eran concluyentes, se procedió a extraer las coordenadas de segmentación para hacerlo sobre la imagen original. Por lo que para proceder a la segmentación de palabras, la cual se extrae a través del algoritmo

Connected-Component Labeling (CCL), se procede de nuevo al pre-procesamiento de las imágenes que contienen las líneas segmentadas.

El pre-procesamiento de las líneas introduce alguna variante al módulo origen de esta herramienta. El proceso de binarización es similar al ejecutado anteriormente, a través de un umbral dinámico se determinan qué píxeles han de tener el valor 0 y cuáles valor 1. La variación se produce a continuación, ya que se procesa la imagen de manera que se genere una dilatación de todos los píxeles negros.

Una vez se dilata la imagen, se ejecuta el algoritmo CCL, que a través de la detección de bordes identificará distintas componentes en la línea. Para determinar qué componentes son identificadas como palabras, y cuales son ruido, se define un umbral dinámico en base a la altura y achura media de las componentes, que excluirá todas aquellas que no sean identificadas como palabras.

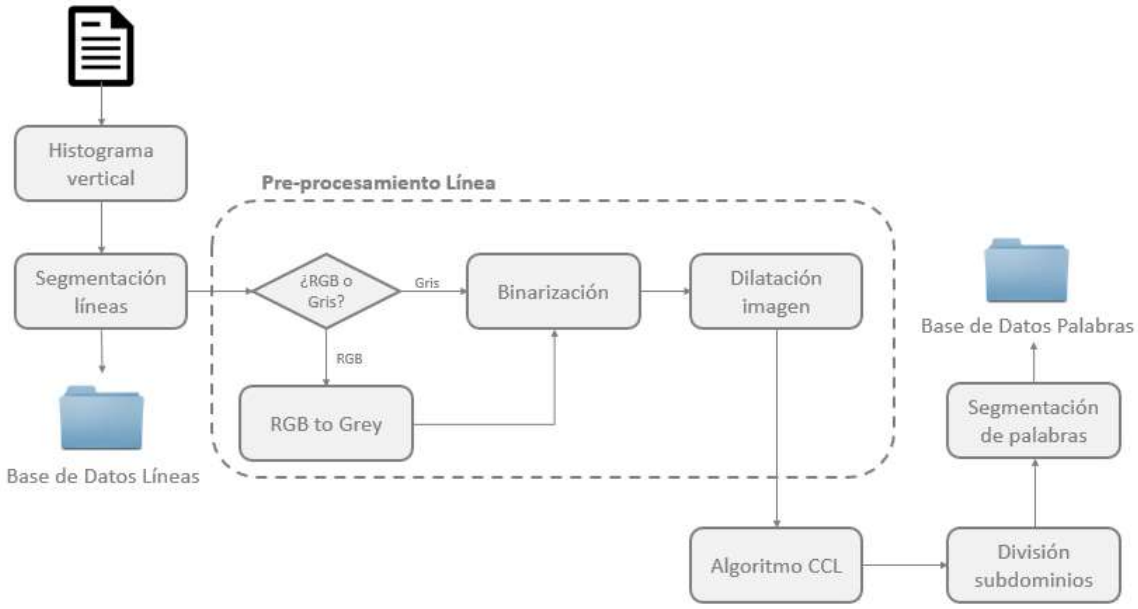

Ilustración 13. Módulo completo segmentación

3.2.3 Bloque 3: Indexación de palabras

El último módulo de la herramienta tiene como función extraer la información de la ubicación de las líneas y palabras segmentadas, con el fin de crear un directorio de trabajo en el que se organicen todos los resultados.

Todas las imágenes extraídas deben estar indexadas haciendo referencia al acta del que han sido extraídas, número de línea y ubicación dentro de la misma. De manera que si un

paleólogo ha de consultar información acerca del contexto de esa palabra tenga una referencia clara que permita redirigir su atención al punto de origen en un corto periodo de tiempo.

Del mismo modo se genera una base de datos donde se recopilan, el número total de líneas y palabras para una fase posterior de pruebas.

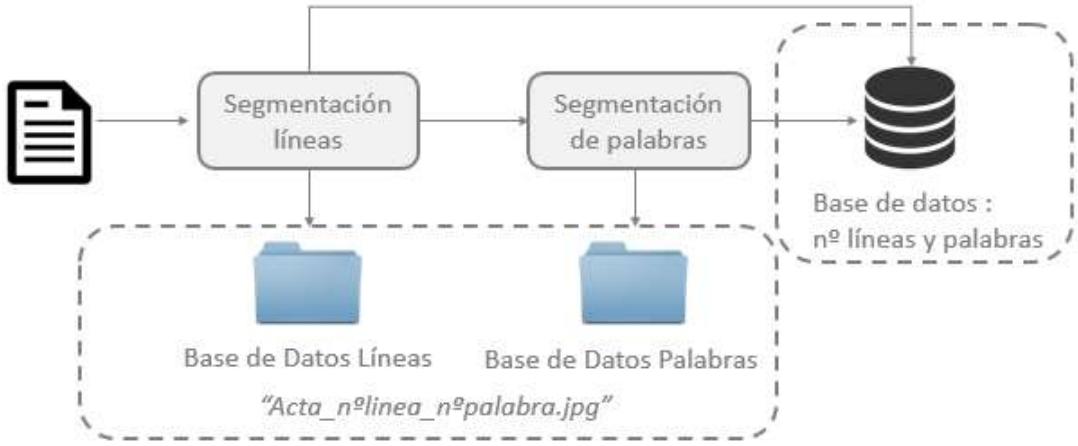

Ilustración 14. Módulo Indexación

4 Desarrollo

4.1 Herramientas utilizadas

Para la elaboración de este TFG se ha utilizado MATLAB [6]. MATLAB es una herramienta de software matemático, que provee un entorno de escritorio afinado para la exploración, el diseño y la solución de problemas de manera iterativa.

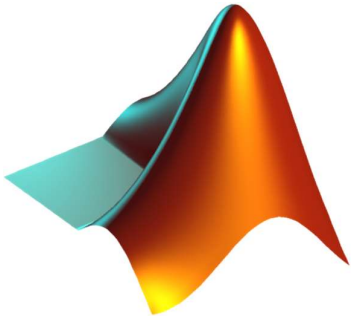

Ilustración 15. Logotipo Matlab

Matlab utiliza su propio lenguaje de programación, lenguaje M. Este traslada cualquier problema un punto de vista matemático. Además posee multitud de órdenes ya configuradas en sus toolboxes que permiten reducir a fracciones de tiempo la resolución de muchos problemas de lo que se tardaría con otros lenguajes de programación.

Otro de sus puntos fuertes, es que es una herramienta muy potente para la visualización de resultados gráficos.

4.2 Implementación

En este apartado se detallará como se ha configurado cada uno de los módulos a nivel de funciones y algoritmos, para trabajar sobre la fuente de origen de trabajo: un subconjunto de las Actas del Puerto de Tarragona del s. XIX.

4.2.1 Pre-procesamiento de la imagen

A continuación se definen las fases en las que se desarrolla el pre-procesamiento de la imagen.

4.2.2 Adquisición de la imagen

Para iniciar la fase de pre-procesamiento se cargan al directorio de trabajo las imágenes del directorio de las Actas de Tarragona en Matlab con ayuda de la función *imread.*
El resultado de esta función será una matriz tridimensional que contendrá un acta en su formato RGB.

Ilustración 16. Variable imagen original

Una imagen de color RGB se representa por tres matrices bidimensionales, correspondientes a los planos rojo, verde y azul para cada píxel. El color se determina con la combinación de los tres planos.

4.2.3 Binarización

La binarización es un proceso que permite reducir notablemente la cantidad de datos de una imagen de una manera muy sencilla, ya que su objetivo es transformar una imagen con distintos niveles de gris, en otra imagen en la que solo estén presentes el blanco y el negro (1 y 0 respectivamente en Matlab).

Para determinar qué píxeles son blancos y cuales negros se define el umbral de binarización. El umbral de binarización es aquel que delimita a partir de qué valor los píxeles van a ser negros y cuales blancos. Todos los valores que estén por debajo de ese umbral, pasarán a ser píxeles blancos (1 en Matlab) y todos los que estén por encima, negros (0 en Matlab).

Dado que los datos con los que se pone a prueba esta herramienta están en formato RBG, ha de convertirse previamente a escala de grises. Para ello Matlab tiene en una de sus toolbox la función *imgrey*. En esta transformación desaparece la información de tono y saturación de la imagen, dejando exclusivamente la luminancia.

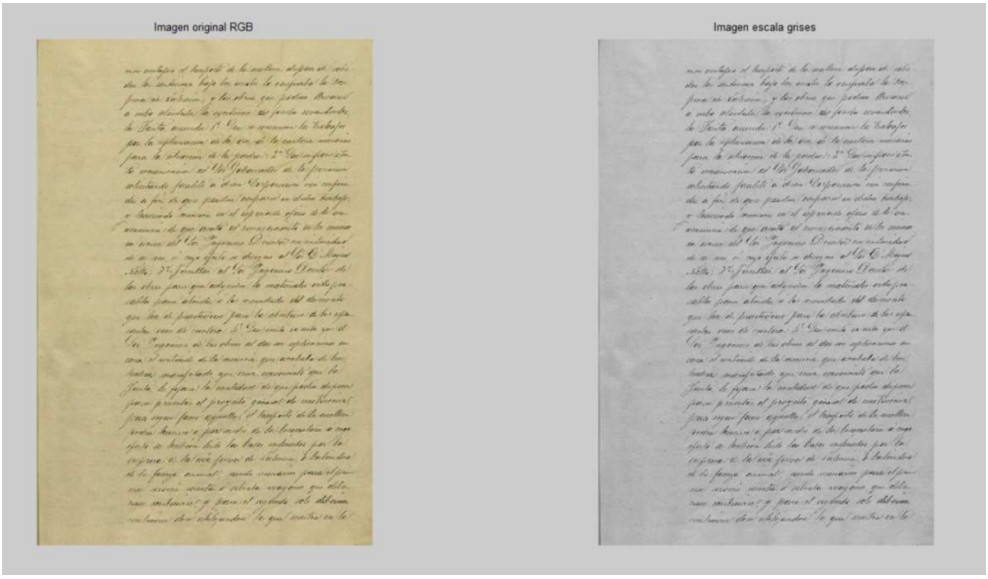

Ilustración 17. Imágenes: original y en escala de grises

Una vez obtenida la imagen en escala de grises, obtenemos la imagen binarizada con la función de Matlab im2bw. Esta función tiene dos parámetros de entrada: por un lado la imagen en escala de grises, y por otro el umbral de binarización.

Para definir el umbral de binarización, se ha utilizado un método denominado umbral Otsu, que determina dinámicamente para cada imagen cual es el umbral más óptimo. Este método proporciona el umbral óptimo para la segmentación de la imagen, bajo el criterio de máxima varianza entre fondo y objeto.

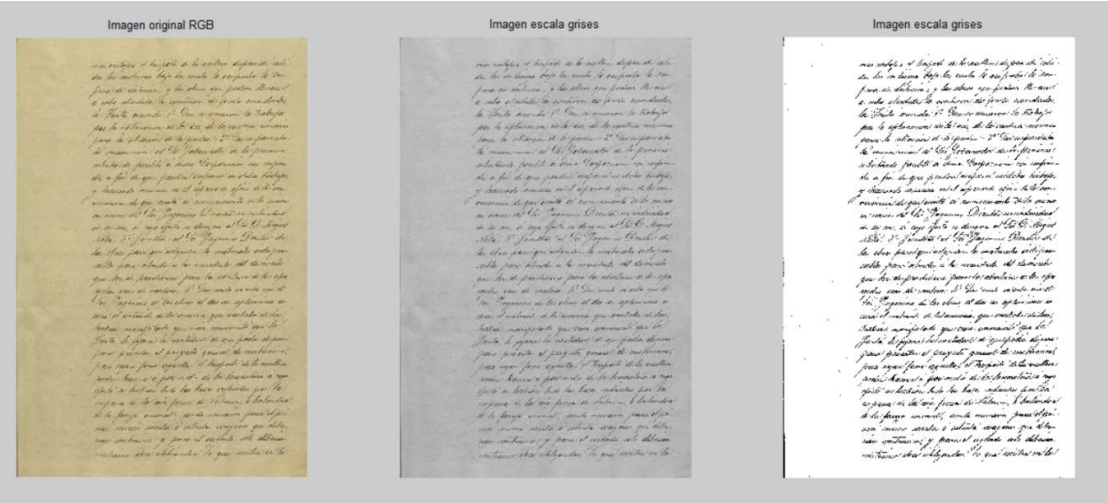

Ilustración 18. Imágenes: original, escala de grises y binarizada

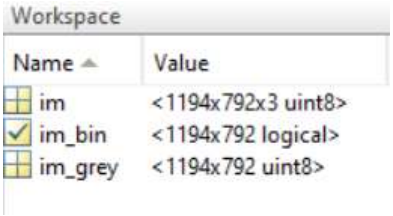

Ilustración 19. Variables de imágenes: original, escala de grises y binarizada

4.2.4 Detección de Slant y Skew

Las páginas en las que se detecte que el ángulo de inclinación no es horizontal, se archivarán para definir soluciones en trabajos futuros. Concretamente para la base de datos sobre la que se realiza el estudio no se han tenido problemas de slant y skew.

4.2.5 Eliminación ruido entre líneas

Para eliminar el ruido entre líneas, que dificultará la segmentación de las mismas, se recorre la imagen verticalmente con el objetivo de generar un histograma donde se vea la concentración de píxeles negros.

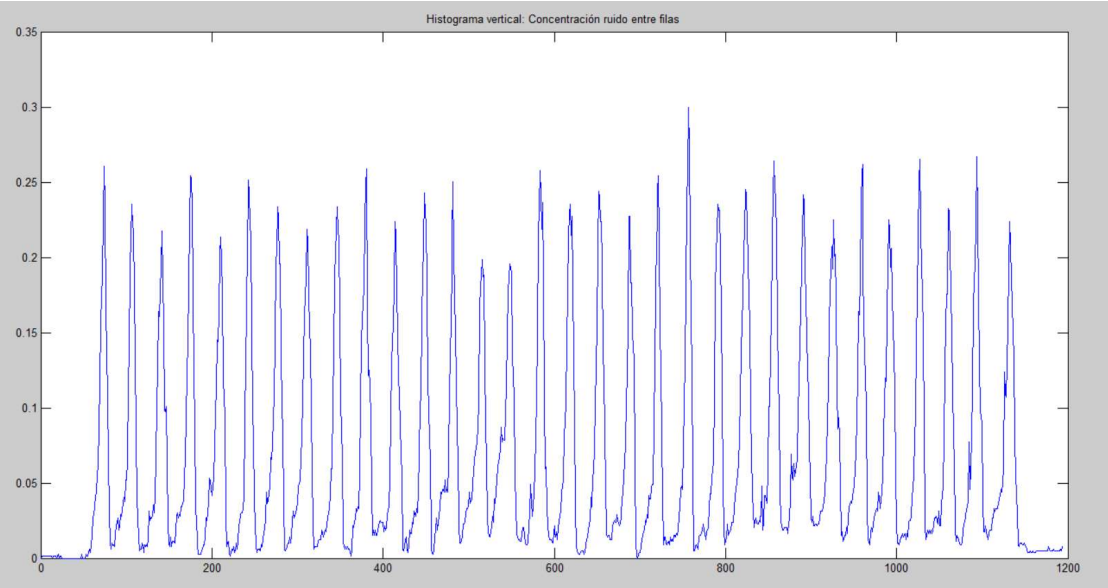

Ilustración 20. Histograma vertical de la imagen binarizada

Por un lado en el histograma se pueden reconocer a simple vista, en qué posición se encuentran las líneas de texto, pero también se puede ver la acumulación de píxeles entre líneas, que sólo introducen ruido a la imagen.

Para la eliminación de ese ruido se determina un umbral, que convertirá en blanco todos aquellos píxeles que estén por debajo del umbral, dejando de este modo la imagen limpia de ruido y preparada para una segmentación a nivel de líneas exitosa.

En primera instancia, la definición de este umbral se intentó calcular en base a distintos estádisticos del histograma (varianza, desviación típica, média,etc). Pero debido a que las imágenes recogidas en estas actas son muy diferentes, no se conseguía un resultado muy óptimo, por lo que finalmente se decidió determinar el umbral manualmente, dejando para trabajos posteriores el diseño de un algoritmo de estimación automática de este umbral.

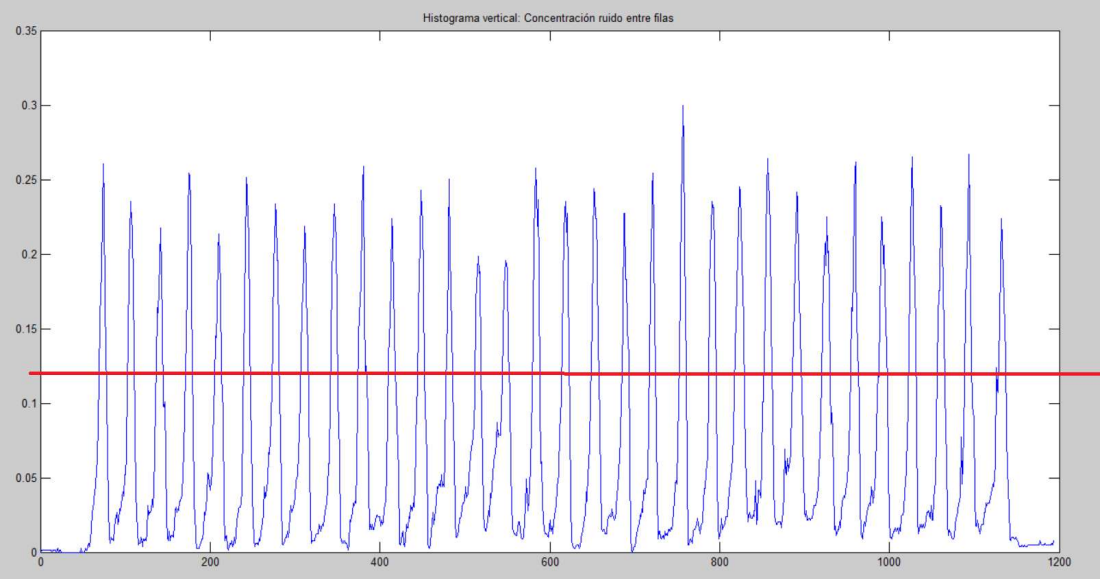

Ilustración 21. Detección de umbral para el filtrado de ruido entre líneas

4.3 Segmentación

4.3.1 Segmentación a nivel de línea

Como ya se ha descrito en el diseño de la herramienta el proceso de segmentación se realiza en primer lugar a nivel de línea, por dos motivos fundamentales:

- La segmentación a nivel de línea ofrece información sobre la ubicación en el texto.
- Permite una segmentación a nivel de palabra más sencilla, ya que el número de píxeles a tratar es más pequeño, y se evita que se solapen componentes entre líneas generando resultados erróneos.

Para la segmentación a nivel de línea, solo ha de aplicarse el filtro para la eliminación de ruido entre líneas, y examinar de nuevo la imagen verticalmente.

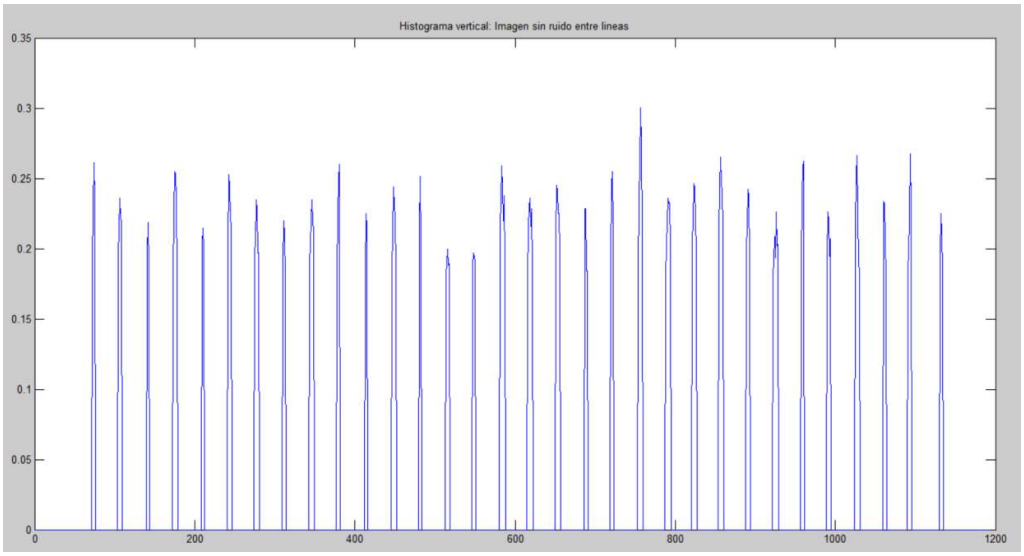

Ilustración 22. Histograma sin ruido

Una vez obtenido el histograma limpio de ruido, se ha de proceder a encontrar la posición en la que comienza y acaba una línea.

Ilustración 23. Ejemplo detección de líneas

Para detectar una línea se recorre la imagen verticalmente, en una variable auxiliar se va almacenando la información de la concentración de píxeles blancos, cuando el algoritmo encuentra un cambio de concentración a píxeles negros guarda esa coordenada como origen de la línea.

Para localizar la coordenada de fin de la imagen, hace el proceso invertido, es decir, recorre la imagen hasta que encuentra el cambio de concentración de píxeles negros a píxeles blancos.

En un origen, se recortaron las líneas tomando esas coordenadas de referencia. Más tarde en el proceso de segmentación de palabras, se pudo comprobar que parte de las palabras aparecían recortadas tanto en la parte inferior como superior, por lo que se decidió afinar el proceso de segmentación, modificando las coordenadas de inicio y fin al punto medio entre el final de una línea y el inicio de otra.

Ilustración 24. Ejemplo línea segmentada

4.3.2 Segmentación a nivel de palabra

La información extraída en la segmentación de palabras, se redirigirá tanto al módulo de indexación como al de segmentación a nivel de palabra.

Este módulo está formado por una función de Matlab *segmentarpalabras.m,* definida para este Trabajo de Fin de Grado, en la que serán procesadas las líneas segmentadas anteriormente, partiendo de la imagen origen.

Por lo que en primer lugar, se realizará de nuevo un pre-procesamiento de las imágenes.

4.3.2.1 Pre-procesamiento de las líneas segmentadas

Al igual que en el módulo anterior, será necesario transformar la imagen de formato RGB a escala de grises a través de la función *rgb2grey.* Una vez convertida la imagen se procederá su binarización, estableciendo el umbral de binarización a través del método Otsu.

Ilustración 25. Imágenes líneas: original, escala de grises y binaria

Teniendo en cuenta que el algoritmo principal de este módulo es el *Component-Connected Labeling,* cuyo objetivo es detectar componentes dentro de una imagen a través de su conectividad, es importante asegurarse de que un mismo componente no va a ser detectado como varios. Esto podría producirse, si existiese por ejemplo, separación entre las letras de una misma palabra. Y por tanto el algoritmo CCL devolvería resultados no válidos. Para anticiparnos a este tipo de error, pasamos la imagen por un proceso de dilatación.

Matlab dispone de una función que hace precisamente esto, y es la que ha sido utilizada en el desarrollo de este proyecto. Esta función es *imdilate.* Esta función tiene dos parámetros de entrada, por un lado la imagen con bordes detectados, y una estructura que ha debido ser definida previamente donde se recogen el tamaño y la forma de la dilatación.

Para la detección de bordes se hace uso de una función de la toolbox de Matlab. Esta función es *edge.* Esta funcíon devuelve una imagen binaria en la que los bordes están reconocidos como píxeles blancos (valor 1), y el resto como negros. Aunque existen diversos operadores para la detección de bordes, en este caso hemos usado Sobel, ya que ha ofrecido buenos resultados.

La detección de bordes es el proceso de localización de las transiciones de intensidad entre los píxeles. El método Sobel calcula el gradiente de la intensidad de una imagen en cada punto (píxel). Así, para cada punto, este operador da la magnitud del mayor cambio posible, la dirección de éste y el sentido desde oscuro a claro. El resultado muestra cómo de abruptamente o suavemente cambia una imagen en cada punto analizado y, en consecuencia, cuán probable es que éste represente un borde en la imagen y, también, la orientación a la que tiende ese borde.

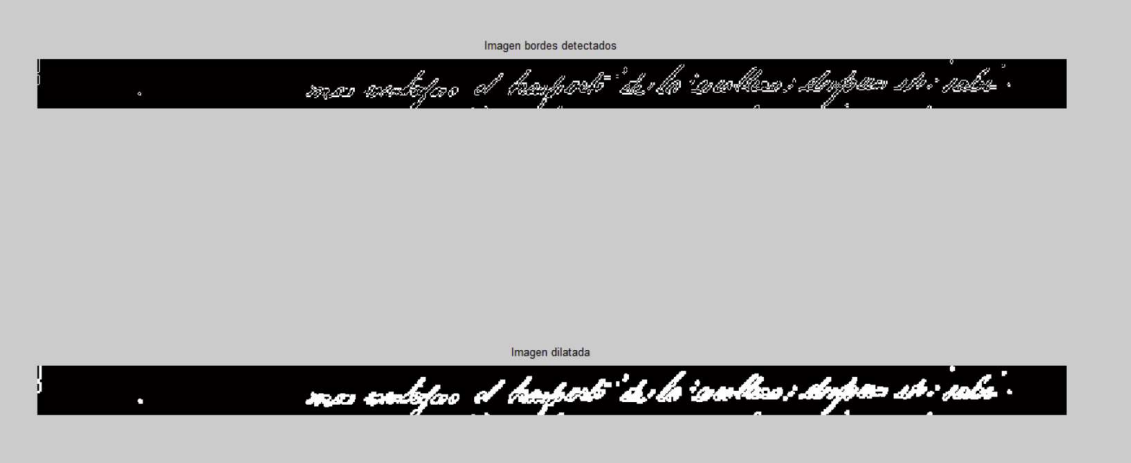

Ilustración 26. Detección de bordes y dilatación de línea

4.3.2.2 Extracción de componentes conexas

Una vez obtengamos una línea dilatada procederemos al reconocimiento de componentes conexas. Para ello haremos uso de la función de Matlab *bwlabel.* La cual asigna una etiqueta a cada componente identificada. Esta función reconoce las componentes en base a la adyacencia de los píxeles.

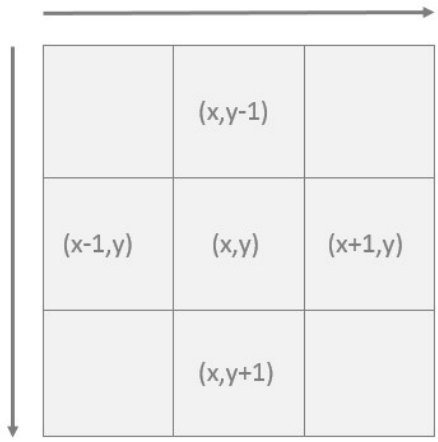

Ilustración 27. Componentes conectadas

Este algoritmo consiste en recorrer la imagen de izquierda a derecha y de arriba abajo siguiendo los siguientes pasos:

1. Durante el primer rastreo, para cada punto P(x,y) que tenga valor 1, se examina a los vecinos superiores A(x-1,y-1), B(x-1,y), C(x-1,y+1) y D(x,y-1); nótese que si existen, acaban de ser visitados por el rastreo, así que si son pixeles negros, ya han sido etiquetados.
    a. Si todos son 0, se asigna P una nueva etiqueta;
    b. Si tan sólo uno es 1 se le asigna a P la etiqueta del otro;
    c. Si hay más de uno que no es 0, se asigna a P la etiqueta de uno de ellos, y si sus etiquetas son diferentes, se registra el hecho de que son equivalentes, y por tanto pertenecen a la misma componente.

2. Cuando se completa el primer rastreo, cada pixel negro tiene una etiqueta, pero puede que se asignen muchas etiquetas diferentes a puntos en el mismo componente.
3. A continuación el algoritmo organiza las parejas equivalentes en clases de equivalencia, y escoge una etiqueta para representar cada clase. Finalmente, realiza un segundo rastreo de la imagen y sustituye cada etiqueta por el representante de cada clase; cada componente ha sido ahora etiquetada de forma única.

La función bwlabel devuelve dos valores, por un lado una matriz, en la que están etiquetados los distintos componentes, y por otro lado el número de componentes dentro de la imagen.

Ilustración 28. Variables Matlab componentes conectadas

A continuación, se hace uso de la función *regioncrops*. Esta devuelve una estructura a la que comúnmente se denomina propied. En la que por cada componente detectada facilita la información acerca del centroide, bounding box y área de la misma.

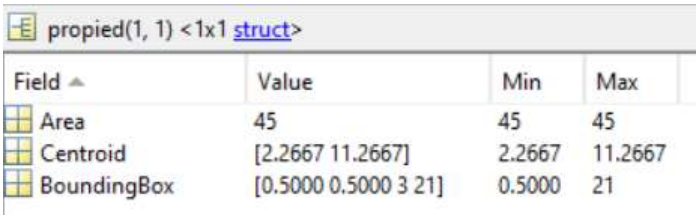

Ilustración 29. Estructura componentes conectadas

En este caso práctico, solo haremos uso de Bounding Box, que nos facilitará las coordenadas de donde está ubicado cada componente.

Bounding Box tiene un formato de este tipo, [xi, yi, x, y]. En el que los dos primeros valores hacen referencia a la coordenada de origen, y los otros dos a altura y anchura del componente, tal y como se muestra en la imagen a continuación:

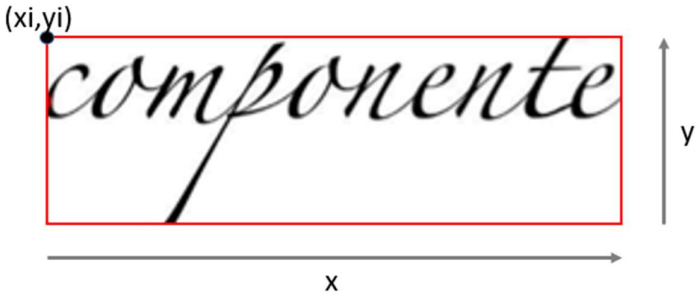

Ilustración 30. Bounding Box

Una vez aplicado el algoritmo CCL a nuestra línea, obtenemos un resultado como el que se muestra a continuación:

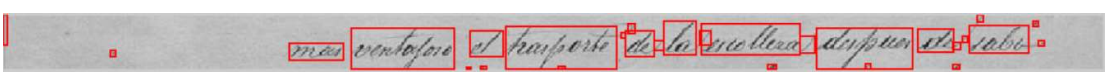

Ilustración 31. Representación componentes conexas

4.3.2.3 Estimación de altura media de los caracteres

Una vez realizado el paso anterior, como se puede comprobar, a primera vista se identifican las palabras, pero también hay otros elementos que no son interesantes, y que por tanto obstaculizan el proceso de segmentación. Para eliminar este problema, se van a organizar las componentes conexas en distintos subdominios, en función de su tamaño.

Para ello, se procede a calcular la altura y anchura media de las componentes. Guardamos en un vector, la altura de todos los componentes; y en otro vector, la anchura. Esta información la podemos obtener de las coordenadas 3 y 4 de *Bounding Box*. Esto se realiza en un proceso iterativo.

```
% Cálculo de ancho y alto de cada componente para crear subdominios.

for n=1:size(propied,1)
    W(n)=(propied(n).BoundingBox(3));
    H(n)=(propied(n).BoundingBox(4));
end

AW=mean(W);
AH=mean(H);
```

Ilustración 32. Extracto de código – Cálculo altura y anchura media

Una vez están definidos los dos vectores, se procede a calcular el valor medio.

4.3.2.4 Definición de subdominios

Como se ha mencionado en el punto anterior, cada componente conexa puede ser un elemento del texto distinto. Esta variabilidad hace necesaria la subdivisión de las componentes en dos subdominios. Los subdominios serán los siguientes:

- Subdominio 1: Contiene a las componentes correspondientes a la mayoría de los caracteres.

- Subdominio 2: Contendrá caracteres tales como acentos, signos de puntuación y pequeños caracteres.

Para definir qué componentes pertenecen a cada subdominio, se establece la siguiente norma:

Todos aquellos componentes que cumplan las siguientes normas, pertenecerán al dominio 2:

- Que la altura de la imagen sea inferior a 3 veces la altura media, y que además la anchura de la imagen no supere la mitad de la anchura media.

- Que la altura de la imagen sea inferior a la mitad de la altura media, y que su anchura tampoco sea superior a la anchura media.

Para todos los demás casos, las componentes se engloban dentro del subdominio 1.

Ilustración 33. Representación subdominios

En la imagen se pueden identificar todos los elementos pertenecientes al subdominio 1, con el rectángulo rojo; y los del subdominio 2 con el rectángulo verde.

Finalmente, y para terminar con el proceso de segmentación, se eliminan todos los componentes del subdominio 2 de la imagen. Gracias a *Bounding Box*, se conoce la ubicación de estos componentes, por tanto se recorre la imagen binarizada, y cuando se encuentra dentro de la zona de algún componente de este subdominio se sustituyen los píxeles, por píxeles en blanco.

Ilustración 34. Segmentación de palabras

4.4 Indexación de palabras

Ya que el objetivo principal de esta herramienta es alimentar un siguiente módulo de clustering.

El algoritmo de clustering permitirá agrupar todas las imágenes (palabras) que tienen un gran parecido. La idea es obtener automáticamente un clúster por palabra y con el centroide de cada clúster (palabra que identifica el clúster) generar un diccionario de imágenes que un especialista debería procesar (transcribir cada palabra-imagen del diccionario a texto). Todas las imágenes de un clúster tendrán asociada el texto de la palabra que representan. En un proceso automático inverso, se generará cada página de un acta sustituyendo cada imagen (indexada) por el texto asociado al centroide del clúster, donde esa imagen estaba incluida. De ese modo, se tendría una transcripción automática textual de cada página de cada acta, utilizando la información de indexado y de la palabra (texto) asociada a cada centroide de cada clúster.

Por tanto, es de vital importancia referenciar correctamente la ubicación de las mismas. De este modo en caso de consulta, siempre existe una referencia en la misma imagen que permita redirigir a la persona que esté trabajando con estos textos al lugar exacto en unos pocos segundos.

Como ya se ha mencionado antes existirán tres referencias:

1. Acta de la que provienen.
2. Línea en la que se encuentra la palabra.
3. Posición de la palabra en la línea.

Para realizar este proceso se ha ido obteniendo información de los bloques anteriores. Por cada vez que se iteraba el procese de segmentación de palabras, se iba guardando en una matriz el número de palabras por cada línea. De igual manera, se iba generando en el directorio de trabajo una base de datos con todos los archivos y correctamente indexados en el título.

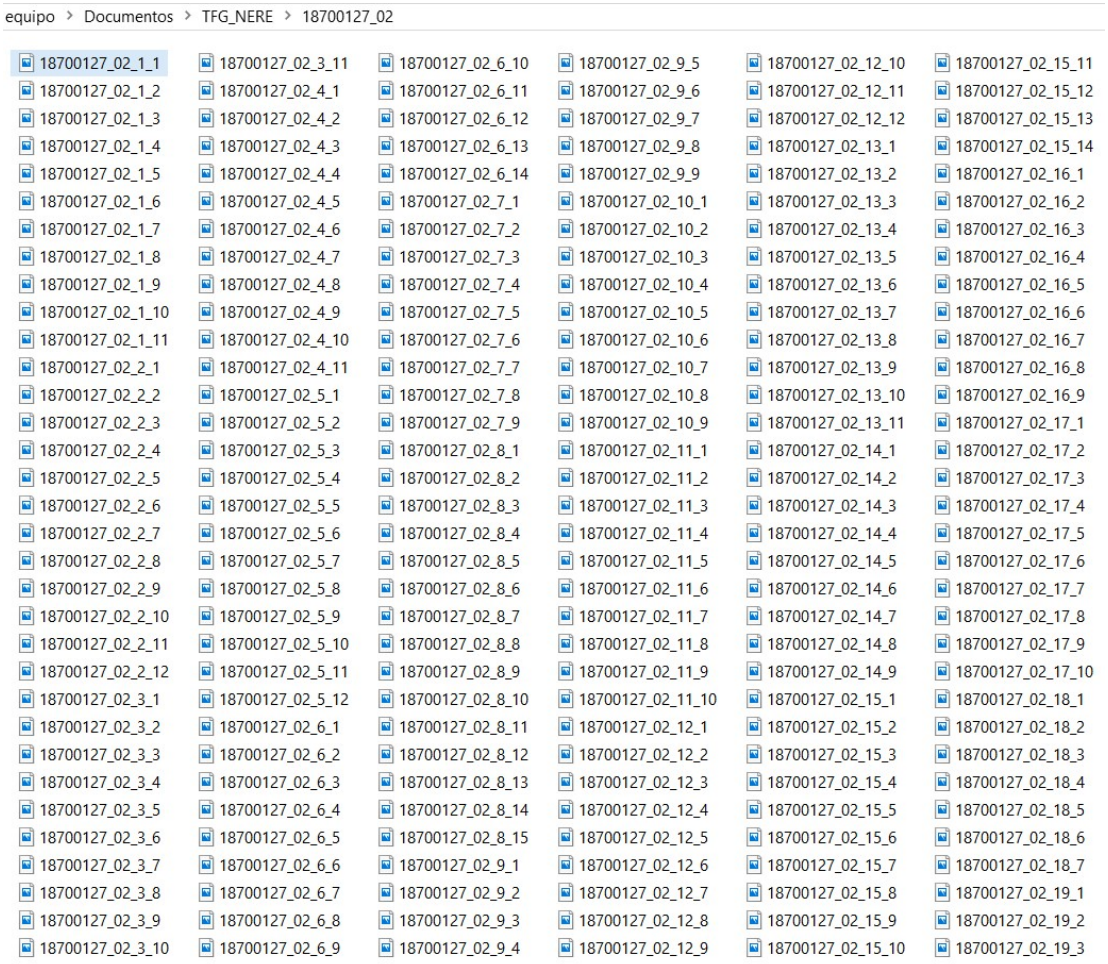

Ilustración 35. Extracto del directorio de trabajo

5 Integración, pruebas y resultados

5.1 Actas del Puerto de Tarragona

Las pruebas de la herramienta desarrollada en este proyecto se han realizado con una base de datos formada por un subconjunto de Actas del Puerto de Tarragona. En Abril de 2010 se iniciaron las tareas de digitalización de los libros de actas de la Junta de Obras del Puerto de Tarragona, una de las series documentales más importantes y ricas en información. Este repositorio se creó con el objetivo de custodiar el patrimonio documental de la administración portuaria desde sus orígenes hasta la actualidad.

El Archivo del Port de Tarragona es un ambicioso proyecto del Port de Tarragona destinado a poner al alcance de los investigadores las series documentales más consultadas o más importantes en formato digital, favoreciendo de este modo la difusión de sus fondos y los estudios alrededor del Port de Tarragona.

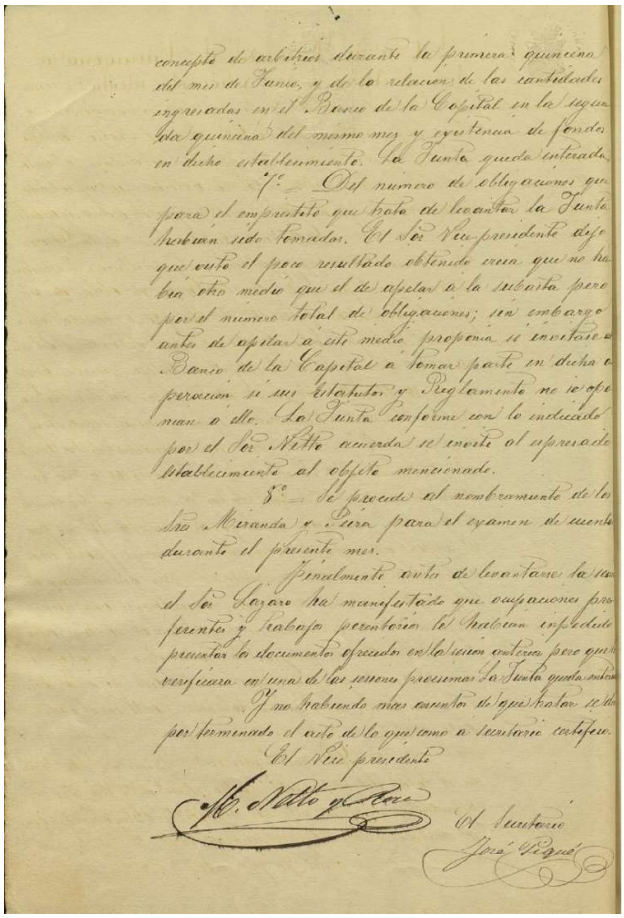

Ilustración 36. Muestra Acta Puerto de Tarragona

La documentación que se ha digitalizado consiste en 22 volúmenes de los libros de Actas del Pleno de la Junta de Obras del Puerto de Tarragona de los años 1869 al 1992 y 20 volúmenes más que corresponden a las sesiones de la Comisión Ejecutiva- Permanente de los años 1903 al 1991. No obstante las empleadas en este trabajo han sido las Actas de la Junta de Obras, del S.XIX.

5.2 Estado de la base de datos

Las páginas escaneadas de las Actas del Puerto de Tarragona presentan múltiples defectos que dificultaron la segmentación correcta de todos los datos.

Uno de los principales problemas, es la calidad del fondo. A lo largo de todas las páginas presenta tonalidades diferentes, aparecen manchas de humedad, incluso se calca la página posterior. Esto supuso que en el proceso de binarización, fuera una tarea complicada, ya que la definición del umbral de binarización dinámico (Otsu) no tuviera la capacidad suficiente como para distinguir entre el texto y el fondo.

Otra de las dificultades presentes es la diferencia de la tipología de las letras, dado que este es un manuscrito. En algunas palabras, la separación entre las letras era lo suficientemente grande como para que el algoritmo reconociera dos componentes dentro del texto.

Ilustración 37. Línea detectada con segmentación de palabras errónea

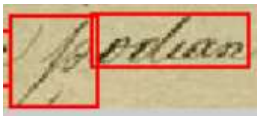

Ilustración 38. Detección de dos palabras errónea

En conclusión, los resultados de la herramienta no son totalmente concluyentes, y por ello abren un mundo de posibilidades a trabajo futuro.

5.3 Resultados Experimentales

Se ha realizado un estudio sobre 25 páginas al azar para establecer la tasa de error que existe con esta herramienta. Para ello se ha realizado un recuento manual del número de líneas y palabras en estas actas, con el fin de comparar con los datos extraídos tras la segmentación.

Los resultados teóricos indican que de promedio aparecen 29 líneas por cada página , y 8 palabras por línea. Los resultados obtenidos tras la experimentación indican que la herramienta ha segmentado un promedio de 30 líneas por página y 11 palabras por línea.

$$P_e = \frac{|Promedio\ teórico\ - Promedio\ experimental|}{Promedio\ teórico}$$

Ilustración 39. Fórmula tasa de error

Por tanto podemos concluir, que la calidad de la segmentación a nivel de líneas es bastante buena, ya que la tasa de error es tan solo del 3,5 %. La aparición de líneas erróneas es un hecho aislado, que podrá ser enmendado en trabajos futuros que tengan como objeto la definición de métodos que realicen el cálculo del umbral de ruido de manera dinámica.

A nivel de segmentación de palabras, la tasa de error se incrementa a un 37.5%. En definitiva, los factores que influyen en la calidad a nivel de segmentación de palabras son múltiples, ya que afectan tanto las características de la propia imagen (ruido, skew, etc), como las irregularidades de la propia caligrafía (separación entre letras, conexión de palabras diferentes, etc…). Los errores provocados por la calidad de la imagen, pueden ser corregidos afinando la fase de preprocesamiento, incluyendo métodos que utilicen umbrales dinámicos, aplicando otra clase de filtros para eliminar el ruido. No obstante, tras la experimentación se ha observado que es probable que para corregir los resultados erróneos provocados por la caligrafía, haya que utilizar otro tipo de método de segmentación.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

El proceso de automatización de segmentación a nivel de palabras es un proceso complejo que ofrece un gran campo de estudio para trabajos posteriores. Tras analizar paso a paso cada uno los módulos que componen esta herramienta, se ha podido comprobar que a pesar de los esfuerzos es muy difícil la automatización total, ya que debido a las diferencias entre las páginas es complicado definir un método que calibre los parámetros de la herramienta y sea válido para todas. Siendo así mucho más complejo para textos de diferente origen.

No obstante, supone un gran avance, ya que los parámetros a definir no son demasiados y automatizan gran parte del proceso, que comparado con lo que existe en la actualidad es un pequeño gran paso. Actualmente el proceso que deben llevar a cabo los paleógrafos para estudiar las obras manuscritas y otros incunables es exclusivamente manual. Concretamente el estudio de las Actas de Tarragona se llevó tres meses y medio en transcribir a un soporte digital con un equipo de 16 personas. El tiempo de ejecución de este proyecto en todas sus fases no supero las 3 horas, por lo que supone una reducción de tiempo considerable.

6.2 Trabajo futuro

Este trabajo ofrece múltiples posibilidades a la hora de plantear trabajo futuro, ya que es un campo no muy trabajado, con muchos detalles aun por pulir. En primer lugar un buen punto de partida para trabajo futuro sería la mejora de la presente herramienta. Podría dedicarse un trabajo al afinamiento de la calibración de los parámetros para que el proceso fuera totalmente automático.

Del mismo modo, podría analizarse esta misma herramienta en otros entornos de programación, que quizás permitan obtener resultados más óptimos, y que incluso permita la creación de una interfaz dinámica que permitiera que personas que son del sector pudiesen poner a funcionar la herramienta y conseguir sus propios resultados.

Otro campo de trabajo, es la búsqueda de otros algoritmos para la segmentación de palabras. Lo ideal sería buscar un algoritmo que fuera agnóstico a la calidad de la imagen y que además pudiera tolerar de origen algunos parámetros como la inclinación de skew y slant.

Finalmente, debe tener presencia en este ámbito el campo de la clusterización, ya que es el paso final para obtener una información de calidad segmentada, referenciada y agrupada. Además este punto supone un gran reto, ya que el proceso de clusterización se parte sin ningún tipo de información acerca de cuantos grupos de palabras similares se van a encontrar.

Referencias

[1] J. C. Galende Díaz, Paleografía y escritura hispánica, Síntesis, 2016.

[2] G. Aubert y P. Kornprobst, Mathematical Problems in Image Processing. Partial Differential Equations and the Calculus of Variations, Springer-Verlag, 2006.

[3] C. A. Cattaneoa, L. I. Larchera, A. I. Ruggerib, A. C. Herrera y E. M. Biasonia, «MÉTODOS DE UMBRALIZACIÓN DE IMÁGENES DIGITALES BASADOS EN ENTROPIA DE SHANNON Y OTROS,» 2011.

[4] Moisés Pastor i Gadea , «Aportaciones al Reconocimiento Automático de Texto Manuscrito,» Valencia, 2007.

[5] G. Louloudis, B. Gatos, I. Pratikakis y K. Halatsis, «A Block-Based Hough Transform Mapping for Text Line Detection in Handwritten Documents,» p. Atenas, 2006.

[6] K. Wong, R. Casey y F. Wahl, Document analysis system, IBM J. Res. Devel..

[7] «Mathworks,» 2017. [En línea]. Available: https://es.mathworks.com/help/. [Último acceso: Febrero 2017].

[8] I. Bar-Yosef, N. Hagbi y K. Kedem, «Line segmentation for degraded handwritten historical documents».